

BSTZ No. 42390P10236
Express Mail No. EL651820814US

UNITED STATES PATENT APPLICATION

FOR

SYSTEM AND METHOD FOR PROTECTED MESSAGING

Inventor:

Luke E. Girard

Prepared by:

Blakely, Sokoloff, Taylor & Zafman LLP
12400 Wilshire Blvd., Suite 700
Los Angeles, California 90025
(310) 207-3800

SYSTEM AND METHOD FOR PROTECTED MESSAGING

BACKGROUND

Field of the Invention

5 The invention relates to the field of computers and passing messages between a pre-operating system program and an operating system present program. More specifically, information may be passed among pre-operating system programs and operating system present programs on a computing device via a protected storage.

Background of the Invention

10 As computers and computing devices are now ubiquitous in our society, computer security issues have become important. Ways of deterring theft of computers and computing devices are evolving to meet the challenges posed by the portable nature of laptop computers, cellular telephones, personal digital assistants, and other computing devices. Various methods of user
15 authentication may be used to provide security and deter theft. These methods include passwords, retinal scan, fingerprint scan, and voice scan.

In some computers, upon powering up, the computer's basic input output system (BIOS) may require authentication such as a password before allowing an operating system to boot. In other computing devices, a
20 password or other authentication must be provided to allow for completion of booting of an operating system, connecting to a network, accessing a database, or starting application programs such as, for example, an electronic mail program. Some programs provide for secure documents such that a document may not be viewed, or edited without entering a password or
25 otherwise authenticating the user's right to access the document.

Although all these security measures exist in various forms, a user may become burdened by and annoyed at having to remember multiple passwords, at having to enter multiple password, and at having to regularly authenticate the user's rights to use the particular computing device, software
30 program, etc.

5

[illegible]

DETAILED DESCRIPTION OF THE INVENTION

Figure 1 illustrates an embodiment of a system architecture in which the system and method for protected messaging are practiced. This invention describes how a new platform primitive, protected storage 130, may be used to send messages between the pre-operating system (Pre-OS) operating space 160 and the operating system present (OS-Present) operating space 170. This protected message passing enables, among other things, the ability to have single, or at least simplified, log on capability. That is, information about the user who logs on to a computing device during booting up and how they logged in may be placed in protected storage so that other Pre-OS programs and OS-Present user applications may access it. Similarly, OS-Present applications may send messages to other OS-Present applications and Pre-OS applications via the protected storage. One example of an OS-Present application leaving a message for a Pre-OS application via the protected storage may be to reconfigure hardware or software components. In this example, a high-level OS-Present application may be used to obtain configuration information that may be used by one or more Pre-OS applications to reconfigure the system. In a related embodiment, an OS-Present application may store an executable routine in the protected storage which is run by a Pre-OS application upon rebooting/restarting. In addition, OS-Present applications may use the protected storage to transfer configuration data, security policies, authentication data and other information among themselves, and may also share this information with or receive this information from Pre-OS applications via the protected storage. Pre-OS applications 100 and OS-Present applications 140 use interfaces 125 and 155 to access protected storage medium 130 to accomplish this method and system. These interfaces are provided by Pre-OS driver 120 and OS-Present driver 150. In this way, a general bi-directional messaging feature is provided.

As used herein, a Pre-OS application program may include a basic input output system (BIOS) program as well as other applications that may execute during boot up before the operating system is loaded and may include applications stored on optional read-only memory (ROM) devices

associated with various peripherals attached to or part of the personal computing device. An OS-Present application program may be any application program that runs while the operating system is present.

In one embodiment, the protected storage medium may be the
5 protected storage hardware or hardware layer of the Intel® Protected Access Architecture (IPAA) described in Application Interface Specification, Rev. 0.9.5 available from Intel Corporation of Santa Clara, California. (the "IPAA Specification"). In this embodiment, interface 125 may be the interface layer described in the IPAA Specification, and Pre-OS driver 120 may be the
10 support layer or service provider described in the IPAA Specification.

Figure 2 illustrates an embodiment of a computing device in which the system and method for protected messaging are practiced. Computing device 200 may be a personal computer, a portable computer, a server, a cellular telephone, a personal digital assistant, a computer tablet, or other computing
15 device. The computing device 200 illustrated in **Figure 2** is of a personal computer embodiment in which processor 210 may execute instructions using main memory 212 which is accessed via memory controller 214. Main memory may be any well known random access memory (RAM) or other volatile memory device. The instructions may be obtained from BIOS chip
20 216 and software stored on disk memory 224 such as operating system 226 and application programs 228. In one embodiment, protected storage 220 may be exclusively coupled to processor 210. In another embodiment, the protected storage may be coupled to the processor via bus 222. In this embodiment, other components may be able to access the protected storage.
25 Instructions may also be provided via drivers 218 which may be included on the BIOS chip 216. Drivers 218 may, in another embodiment, be included as part of the protected storage medium. It is the drivers that provide the interfaces between the Pre-OS applications and the protected storage, and between the OS-Present applications and the protected storage. In one
30 embodiment, the drivers include a Pre-OS driver and an OS-Present driver to provide interfaces that enable applications to access the protected storage.

Disk memory 224, modem 230 and graphics controller 232 may be coupled to processor 210 via bus 222. Disk memory 224 may be a hard disk

drive, a readable and writeable compact disk (CDRW) drive, a floppy disk drive, a stick or card memory device, a digital audio tape (DAT) reader, etc., or any storage device or other machine readable medium local to the processor, as well as connected by a network or any method of

5 communication, including, for example, wireless. In various embodiments, disk memory 224 may be any device by which a machine may read from a machine readable medium known to those skilled in the art, including all forms of optical and magnetic disks, tapes and similar media; ROM, RAM, and similar memory devices; etc. Processor 210 may display images on
10 display 234 via graphics controller 232. All of the components within computing device 200 are well known and will not be discussed in detail so as not to obscure the subject matter of the present invention.

Protected storage 220 may be any non-volatile readable and writeable memory device, such as, for example, magnetic storage media including hard
15 disks, optical storage media including CDRW, flash memory devices, stick memory devices, and the like. In one embodiment, the protected storage is permanent to the computing device and may not be easily removed. The level of protection shared or accessible from the protected storage is determined by the underlying protected storage technology. The underlying
20 storage technology determines the protection mechanism, protocol, and other security requirements which can be known *a priori* by both Pre-OS and OS-Present applications or, in another embodiment, negotiated during a handshake process. In one embodiment, the negotiations may occur during installation or set up of a particular application program.

25 **Figure 3** illustrates a flow of actions taken according to an embodiment of a system and method for protecting messaging. On a particular computing device a method of the present invention may include providing a protected storage medium, as shown in block 310. To make the storage medium usable, the method includes providing a first interface to the protected storage
30 medium to enable a Pre-OS software program access to the protected storage medium, as shown in block 320. Similarly, a second interface to the protected storage medium is provided to enable an OS-Present software program access to the protected storage medium, as shown in block 330. By virtue of

providing these interfaces, the Pre-OS software program is enabled to securely pass an information to a second Pre-OS software program via the protected storage medium, as shown in block 340. Similarly, the OS-Present software program is enabled to securely pass an information to a second OS-Present software program via the protected storage medium. Further, the method enables the Pre-OS software program to securely pass an information to the OS-Present software program via the protected storage medium, as shown in block 360. In addition, the OS-Present software program is enabled to securely pass an information to the Pre-OS software program, as shown in block 370. Although only one Pre-OS application program and one OS-Present application program are discussed regarding **Figure 3**, it is contemplated that multiple Pre-OS application programs and OS-Present application programs may use the protected storage for secure message passing among themselves, as shown in **Figure 1**, and as discussed below.

In one embodiment, the computing device may have a certain policy which determines what is required for boot or logon to continue. In some embodiments, there may be single factors such as a single password, or, in other embodiments, there may be multiple factors such as two or more of a password, a retinal scan, a fingerprint scan, a voice print identification, location of logon such as an Internet Protocol (I.P.) address, a smart card scan, etc. The Pre-OS applications on a particular computing device may have weaker or less numerous requirements than OS-Present applications, and some OS-Present applications may have more stringent requirements than others. Protected storage may be used to store information about both how the identity of a user was determined and how the user was authorized so that particular applications or the operating system may make a determination if one or more additional authentication measures are required or if access should be denied. In this context, a user may be a human user of a computing device or may be an application program. By allowing Pre-OS and OS-Present application programs to pass information such as authentication information among themselves via the protected storage, a user's computing experience may be made more rewarding.

For example, a Pre-OS application such as a BIOS program may require that the user type in a password as authentication information. The BIOS may then store this information in the protected storage. A later executing Pre-OS application program may access this password information or a
5 message from the BIOS that the user was authenticated by receipt of a password. Based on receipt of this authentication information, the later executing Pre-OS application program may choose not to request a typed in password. The same may apply for OS-Present application programs. Another Pre-OS application or an OS-Present application may obtain further
10 authentication information from a user and either store the authentication information in the protected storage or store an information specifically directed to another OS-Present application. The information passed may be the specific authentication information or may be a notice stating whether the authentication was successful. In this way, later executing Pre-OS and OS-
15 Present applications will not need to re-authenticate the user. In this way, later executing Pre-OS and OS-Present applications may use earlier obtained authentication information from the protected storage to either alleviate the need to further authenticate or reduce the extent of later authentication measures. For example after receiving a password, a later application may
20 not seek a password from the user and may only request the sliding of a smart card or the presentation of a biometric means of authentication such as voice print, retinal scan and fingerprint scan.

This same communication mechanism may be used to send messages in the other direction as well, that is, from the OS-Present space to the Pre-OS
25 space. Passing messages from the OS-Present space to the Pre-OS space may be useful when an OS-Present application wants the Pre-OS application to do something on its behalf, but would not want the action to occur while the operating system was present due to security or other considerations. In this embodiment, when a computing device boots, or rather, restarts or reboots, it
30 may check an assigned or designated protected messaging area of the non-volatile protected storage for messages. In one embodiment, a particular Pre-OS application may be the only software allowed access to a designated protected messaging area. In one embodiment, the protected storage may

include a plurality of cells or slots. In this embodiment, a set of cells or slots may be designated solely for access by particular applications. In this embodiment, the kind of allowed access may vary for each area of protected storage, and may include, read, write, read once, write once, etc. In one embodiment, the information may be designated as read once, auto-erase, so that the data retrieved is then immediately erased from the protected storage. This adds another dimension of security to the protected storage. The Pre-OS application may check that the message placed by an OS-Present application into the protected storage and retrieved by the Pre-OS application was authentic, unaltered, and subsequently “trusted.” Based on the information retrieved, the Pre-OS application may take any requested or appropriate actions.

The kind of information written to and retrieved from the protected storage will vary based on the purpose and use of the information, and the writer and recipient of the information. The type of information that needs to go into the special protected storage “cell/slot” is determined by the applications that will consume it. But there will be common pieces of information that may be required. Information that may be stored in the protected storage or may be made available by the protected storage include, for example: a requested action data specifying what action is being requested; an identity data identifying the user or application program that is requesting an action; an identity credential data which may attest to the identity of the user or application program; a policy data reporting the policy from the requestor’s perspective; a policy credential data that validates the requestor’s policy; a completed action data stating what action(s) has/have been accomplished; a miscellaneous-opaque data that may be use specific and may allow for extensibility and customization; and an integrity data that may be used to ensure that a message has not been altered. In addition, various other kinds and forms of data may be accessed via the protected storage depending on the use and purpose of the data and the purpose and goal of the particular application program.

By adding protected storage to a personal computing device with Pre-OS and OS-Present application programs may access protected storage,

